分散式角色激活管理中的角色查找

林 炼，黎忠文
（厦门大学信息科学与技术学院 福建 厦门 361005）

【摘 要】：角色激活是基于角色的访问控制中的重要环节，用户通过激活系统为他分配的角色子集来执行与角色相对应的权限。目前基于角色访问控制主要特点是用户主动、系统被动，从而给用户带来记忆各种角色-权限分配情况的负担。本文提出一种分散式的基于查找的角色激活管理方法，与传统方法不同之处在于它是一种用户为被动、系统为主动的智能访问控制，在对企业环境下的动态约束提供有效支持外，减轻了在传统角色激活方式下用户需要掌握角色-权限分配情况的负担。

【关键词】：分散式结构 基于角色的访问控制 角色激活

一、引言

近年来，基于角色的访问控制（role-based access control,简称RBAC）已经被公认为较适合在大型企业的计算机网络中实施的访问控制技术。具有代表性的RBAC模型有Sandhu等人提出的RBAC96及其补充模型[1,3]、ARBAC97[4]、ARABC02[5]和CL03[6]等。

Richard W.C. Lui等人在2005年提出了RBAC中的角色激活管理[7]，主要有三种方式：自主式、集中式和分散式。无论采用哪种角色激活管理方式，当前的RBAC都是用户主动、系统被动的模式，它要求用户知道复杂的角色-权限分配状况，这对用户来说是个很大的负担，也使用户倾向于在每次登录时激活最大权限的角色，违背了最小权限原则（PLP），对系统的安全性是个很大的威胁。Raman Adaikkalavan和Sharma Chakravarthy在2006年第一次提出了基于查找的角色激活的概念[8]，为上述问题的解决提出了一个好的思路。基于查找的角色激活根据用户的访问请求，通知用户激活拥有适当权限的角色。

本文以分散式结构为背景，提出了基于角色查找的智能访问控制机制，在利用了分散式结构的优点的同时避免了用户必须记忆繁琐的角色-权限分配的缺点。目前我们尚未查到相似的研究。

二、分散式角色激活管理

NIST RBAC Standard[9]定义了RBAC的四个组件，分别为：

● 核心RBAC：包括用户、角色、权限以及它们之间的关系。

● 角色的层次关系：角色之间的层次关系在数学上描述为偏序关系，其中高层角色继承低层角色的权限，底层角色继承高层角色的用户。

● 静态职责分离（Static separation of duty, SoD）：通过用户-角色（U-R）分配避免用户获得互斥角色。

● 动态职责分离（Dynamic SoD）：可以为用户分配互斥角色，但在角色激活时根据系统的安全策略对互斥角色的激活进行限制。

角色激活服务（role activation service, RAS）根据角色激活策略（role activation policy, RAP）以及用户对应的激活角色集（activated role set, ARS）来决定是否允许某角色的激活。

在分散式角色激活管理的模型（如图1）中，RAP在不同位置由不同的域安全管理员定义。资源（数据和应用）存储于多个服务器中，服务器位于不同的域或子域。资源对应的P-R分配也由相应的安全管理员定义。每个域有自己的RAS来处理用户的角色激活请求。

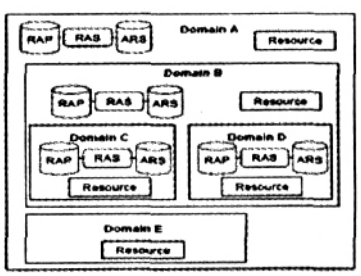

图1 分散式角色激活管理模型

在某个域中，RAS执行该域的RAP以及祖先域的RAP，并结合该域的ARS来决定是否接收用户提交的角色激活请求。

分散式的角色激活提供了描述动态约束的灵活性，将安全策略分散到各个域中，这与现实生活中各个域的安全管理员对所在域的安全策略比较熟悉的现象是吻合的。该模型还提供了一定的容错能力，当一个域的RAS出错或受到攻击时，其它域的RAS仍然可以正常工作。

三、将角色查找运用于分散式结构中

Raman Adaikkalavan等提出的角色查找算法[8]中，没有考虑角色的层次关系以及域的层次关系，并不适用于分散式结构。本节根据分散式结构[7]的要求，对角色查找算法进行改进。改进后的算法（如图2）基于分散式结构，将更适应于大型企业环境。系统中定义的各个实体如下：

基于查找的分散式访问请求检测

```
1   INPUT: Session, operation, object
2   OUTPUT: result: BOOLEAN, role R_k
3   session ∈ S; operation ∈ OPS; object ∈ OBJS; PORS
4   Retrieve all active roles in session as ACRS
5   Retrieve all assigned roles for session as ASRS
6   foreach role R_i ∈ ACRS do
7       If (operation, object, ∀R_m ∈ Junior(R_i)) ∈ PR then
8           return TRUE /* allow object access request */
9   foreach role R_j ∈ (ASRS − ACRS) do
10      R_lp = R_j      /* initialize R_lp as R_j */
11      foreach role R_m ∈ junior(R_j) do
12          If (operation, object, R_m) ∈ PR and R_m < R_lp then
13          /* check RAP in domain d=OD(object), and RAP in d's ancestor domain */
14              If rap(d=OD(object), R_m) = TURE
15              and rap (ancestor (d), R_m) = TURE and dynCon(R_m) then
16                  R_lp = R_m
17  PORS = PORS ∪ R_lp
18  /* check if there is a role that can be activated */
19  if PORS ≠ ∅ then
20      return R_k = plp(PORS)   /* output roles that can be activated */
21  else
22      return FALSE             /* deny object access request */
```

图2 基于查找的分散式访问请求检测算法

基金项目：中山市科技项目（2006A157）；厦门大学院士基金（0630-E23011）；厦门大学新世纪优秀人才基金（0000-X07116）。

用户集 U、客体集 OBS、操作集 OPS、权限集 $P = 2^{OBS \times OPS}$、角色集 R、域集 D

角色层次关系 $RH \in R \times R$，是 R 上的偏序关系

域层次关系 $DH \in D \times D$，是 D 上的偏序关系

用户到角色的多对多分配 $UR \in U \times R$

权限到角色的多对多分配 $PR \in P \times R$

对象到域的映射关系 $OD: O \rightarrow D$

域 d 中的冲突角色对集 $CDR \subseteq 2^{R \times R}$

用户到激活角色集的映射关系 $ACRS_d: U \rightarrow 2^R$

用户到系统为用户分配的角色集的映射关系 $ASRS: U \rightarrow 2^R$

用户到冲突角色集的映射关系：$CRS_d: U \rightarrow 2^R$

在用户分配的角色中，具有访问权限但未被用户激活的角色集合 PORS

1、2 行是输入和输出。第 3 行描述了包含输入、角色的集合。

4、5 行检索得到用户已激活的角色集 ACRS 以及用户分配的角色集 ASRS。

6 至 8 行检索用户已激活角色中是否有访问权限，如果有足够权限，则接受该次访问；由于引入了角色层次，这里除了要检测用户已激活角色本身外，还需检测在角色层次关系中的低层角色，其中 junior(Ri) 函数获得比 Ri 低层的角色集。

9 至 17 行对每个为用户分配且未被激活的角色 $R_j \in ASRS - ACRS$ 进行检测，检测过程从 R_j 以及比 R_j 低层的角色中找出：具有尽可能小访问权限的且满足该域 RAP、祖先域 RAP 和其他动态约束（如时间约束，算法中用 $dynCon(R_j)$ 统一表示）的角色，并将这个角色加入到 PORS 中。其中函数 rap (d = OD(object, R_j) 判断 R_j 是否满足域 d 的 RAP。然后把能通过系统 RAP 的角色中加入集合 PORS 中。

18 至 22 行判断 PORS 是否为空，如果不空，系统通过函数 plp (PORS) 获得集合 PORS 中具有最小权限的角色并通知用户，否则，系统将拒绝该次访问请求。

四、算法的分析

算法包括了 RBAC 的四个组件，其中静态 SoD 是由域安全管理员定义的，所以算法中没有体现。

信息泄漏：算法中查找的对象均是在系统为用户分配的角色范围内，用户不会获得除了他自身相关角色以外的信息，保证了在查找过程中没有出现信息泄漏的情况。

查找结果：Raman Adaikkalavan 等提出算法[8]将第一个查找到的有足够权限的角色或所有拥有足够权限的角色集通知给用户，用户还是可能激活拥有过大权限的角色。本文的算法根据系统定义的角色层次关系筛选出尽可能小权限的角色通知给用户。后者比前者增加了筛选步骤，增加了系统的响应时间，但得到的角色的权限是最小的。

域及其子域的 RAP：分散式结构中，子域 RAP 需要服从祖先域的 RAP。企业通过这种子域与祖先域的关系，制定统一的安全策略。Richard W.C. Lui 等人在文章[7]中并没有指出子域如何执行其祖先域的 RAP。我们认为视系统的安全策略可以分为两种方式，方式一：子域将祖先域的 RAP 的执行请求提交给祖先域的 ARS，让祖先域的 ARS 执行其 RAP；方式二：祖先域在每次修改其 RAP 后，主动下发给其所有子域。

方式一可以隐藏祖先域的安全策略，子域只知道执行结果，但不知道祖先域的安全策略的细节，通过这种方式可以保护企业高层的决策。同时，由于激活每次请求都要经过祖先域，这将降低系统的响应速度，并增加祖先域的 ARS 的负担，使之成为系统的瓶颈。方式二的优缺点正好与方式一相反。因此具体应用时要看企业需要。

五、结论

角色激活是基于角色的访问控制中的重要环节。分散式的角色激活管理有着灵活描述动态约束的优点，而基于查找的角色激活使用户只关注所要访问的对象，而无需掌握经常变化的角色-权限分配关系。本文将两者结合，实现了基于查找的分散式角色激活管理方式，利用了两者的优点，使访问控制更切合于实际。文章中针对的是分层次的域结构，实现的算法也是针对这样的域结构，然而，实际中存在不同的域结构，我们将在后期的工作中继续研究。

参考文献：

1.Sandhu R, Coyne EJ, Feinstein HL, Youman CE. Role-Based Access Control Models. IEEE Computer, 1996,29(2):38?47.

2.Sandhu R. Rationale for the RBAC96 Family of Access Control Models. In: Youman C, Sandhu R, Coyne E, eds. Proc. of the 1st ACM Workshop on Role-Based Access Control. New York: ACM Press, 1996. 38?47.

3.Hong F, He XB, Xu ZY. Role-Based Access Control. Mini-micro System, 2000,21(2):198?200 (in Chinese with English abstract).

4.Sandhu R, Bhamidipati V, Munawer Q. The ARBAC97 model for role-based administration of roles. ACM Trans. on Information and Systems Security (TISSEC), 1999.2(1):105?135.

5.Oh S, Sandhu R. A model for role administration using organization structure. In: Sandhu R, Bertino E, eds. Proc. of the 6th ACM Symp. on Access Control Models and Technologies (SACMAT 2002). Monterey: ACM Press, 2002. 155?162.

6.Crampton J, Loizou G. Administrative scope: A foundation for Role-Based Administrative Models. ACM Trans. on Information and System Security (TISSEC), 2003,6(2):201?231.

7.Richard W. C. Lui, Sherman S. M. Chow, Lucas Chi Kwong Hui, Siu-Ming Yiu: Role Activation Management in Role Based Access Control. ACISP 2005: 358-369

8.Raman Adaikkalavan, and Sharma Chakravarthy: Discovery-Based Role Activations in Role-Based Access Control , Workshop on Information Assurance, in Proc. of The 25th IEEE International Performance Computing and Communications Conference Engineering (IPCCC), April 10-12, 2006, Phoenix, Arizona, USA.

9.RBAC Standard, ANSI INCITS 359-2004, InterNational Committee for Information Technology Standards, 2004.

10.杨秋伟，洪帆，杨木祥，朱贤。基于角色访问控制管理模型的安全性分析。Journal of Software, Vol.17, No.8, August 2006, pp.1804?1810

（上接第 1 页）

4.王旭，王宏，王文辉.人工神经元网络原理与应用[M].沈阳：东北大学出版社，2000.

5.许东，吴铮.基于 MATLAB6.X 的系统分析与设计-神经网络[M].西安：电子科技大学出版社，2002.

6.孙全力，王玉兰.改进 SOFM 网络算法及其在裂隙统计分析中的应用[J].地质灾害与环境保护，2003(2).

7.王升明，李淼.一种基于改进的自组织特征映射网络的文档聚类方法[J].计算机工程与应用，2005(3).

8.陈惠兵，夏辉，刘高辉.基于改进 SOFM 网络的调制方式的自动识别[J].应用科技，2006(4).